1973
USDA FOREST SERVICE RESEARCH PAPER PNW-157

A Computer Program for Determining the Load-Carrying Capability of the Running Skyline

Ward W. Carson
and
Donald D. Studier

PACIFIC NORTHWEST FOREST AND RANGE EXPERIMENT STATION
U.S. DEPARTMENT OF AGRICULTURE FOREST SERVICE
PORTLAND, OREGON

ABSTRACT

The running skyline is a popular cable logging system being used in many parts of North America today. Proper application of this system requires investigation of its adaptability to the terrain being logged. This paper presents a digital computer program to determine running skyline load-carrying capabilities to aid the logging layout designer with this planning task. The Fortran IV program, the details of input, and the interpretation of output are discussed.

KEYWORDS: Logging, computer program, forest cutting systems.

WARD W. CARSON is principal mechanical engineer, Pacific Northwest Forest and Range Experiment Station, Seattle, Wash.

DONALD D. STUDIER was civil engineer, Pacific Northwest Forest and Range Experiment Station, Seattle, Wash. He is now with Region 6 of the U.S. Forest Service, Portland, Oreg.

CONTENTS

INTRODUCTION

The running skyline is a cable system widely used for logging in the timber harvesting industry. It is a system of two or more suspended, moving lines, generally referred to as a main and haulback, that, when properly tensioned, will provide lift and travel to the carriage. The basic system is illustrated in figure 1.

A logging system designer must know what payload a logging system can carry over a given ground profile. Recognizing this requirement, Mann[1/] presented a procedure for determining the load-carrying capability of the running skyline and discussed the mechanics of some of the system's configurations. Mann's procedure was an extension of the graphical-tabular method discussed in the "Skyline Tension and Deflection Handbook."[2/]

Mann's procedure provides a straightforward approach to determination of payload capability of the running skyline at midspan. Unfortunately, the method can be time-consuming, generally proceeding at a rate of 10 to 20 payload determinations per man-day. This rate may not be acceptable when a large number of skyline roads are to be designed. A more efficient approach employs a desk-top computer/plotter.[3/] However,

[1/] Charles N. Mann. Mechanics of running skylines. USDA For. Serv. Res. Pap. PNW-75, 11 p., illus., 1969. Pac. Northwest For. & Range Exp. Stn., Portland, Oreg.

[2/] Hilton H. Lysons and Charles N. Mann. Skyline tension and deflection handbook. USDA For. Serv. Res. Pap. PNW-39, 41 p., illus., 1967. Pac. Northwest For. & Range Exp. Stn., Portland, Oreg.

[3/] Ward W. Carson, Donald D. Studier, and Hilton H. Lysons. Running skyline design with desk-top computer/plotter. USDA For. Serv. Res. Note PNW-153, 21 p., illus., 1971. Pac. Northwest For. & Range Exp. Stn., Portland, Oreg.

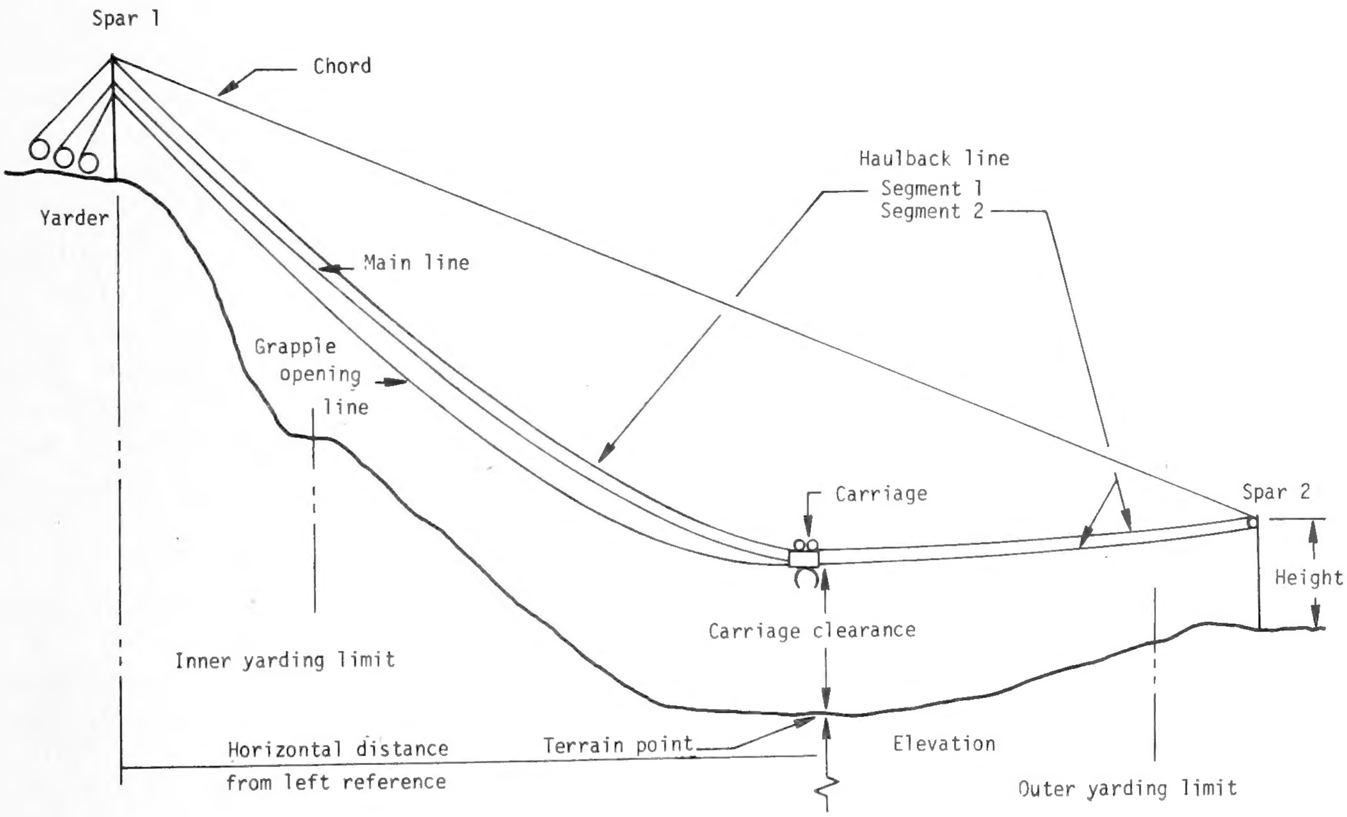


Figure 1.--*Running skyline.*

when the layouts involve many skyline roads and desk-top facilities are not available, it is expeditious to employ a high-speed digital computer for direct computation of payload capability. A computer program designed for this purpose is presented and discussed in this paper.

This computer program is an analytical tool that can be used by the timber resource planner for determining the feasibility of the running skyline system. It has been prepared to provide the logging systems designer with a tool for computing the vertical load-carrying capability of a grapple-rigged running skyline. The program requires information about the ground profile over which the system is operating, the equipment used, and the tension to be maintained in the haulback line during the yarding process. The method of solution is essentially that discussed by Carson and Mann.[4/] The program was written in standard Fortran IV language for the CDC 6400 machine. It has also been prepared to operate on the CDC 3100. An attempt has been made to keep the language standard enough that it will operate on most machines having a Fortran IV compiler.

PROGRAM LIMITATIONS

The designer using this program should be aware of some of the limitations of the results. The program has been designed to compute the load-carrying capability of a grapple-rigged running skyline system. These computations assume that an operating tension exists in the haulback line, that the load is suspended vertically below the carriage, and that the other force required to hold the log and carriage in position is provided by the main line only. The main line tension is limited to a maximum value; therefore, either the haulback operating tension or the main line maximum tension may decide the load-carrying capability.

In an actual situation, it is quite possible that one end of the log will be dragging on the ground during the yarding process. In this case, the full weight of the log will not be felt by the running skyline system. This case differs in two significant ways from the situation where the log is suspended clear of the ground and hangs directly below the carriage: the vertical force on the carriage is less than the log weight, and there is a horizontal force on the carriage due to the log dragging. This case is not treated by the program discussed in this paper. For such situations, this program can only provide the designer with some quantitative feel for the system's capability.

Caution must be exercised in applying the results of this program to a running skyline system which uses a slack-pulling carriage. When a slack-pulling carriage is used, it is possible to design the equipment so that the yarding force is shared by both the main line and the slack-pulling line. Since this program assumes that only the main line bears this force, it does not apply in a shared situation. In those cases where the main line does bear the force and the log is hanging below the carriage, the results are applicable.

Therefore, caution should be exercised when the results of this program are used for any running skyline configuration other than the grapple carriage supporting a completely suspended load, or a slack-pulling carriage supporting a suspended load with negligible tension in the slack-pulling line. When other situations exist, this program is suggested merely as a

[4/] Ward W. Carson and Charles N. Mann. A technique for the solution of skyline catenary equations. USDA For. Serv. Res. Pap. PNW-110, 18 p., illus., 1970. Pac. Northwest For. & Range Exp. Stn., Portland, Oreg.

tool to give the designer a conservative estimate for the capabilities of the running skyline system.

PROGRAM DESCRIPTION

The basic data needed as input for the grapple-rigged running skyline program consist of equipment specifications and individual skyline road specifications. More than one set of skyline road specifications may be used with each set of equipment specifications.

The equipment specifications include the weight of the carriage; the carriage clearance above ground; the headspar and tailspar heights; and the diameter, weight per foot, breaking strength, and safety factor of the haulback and main line cables. The weight and breaking strength for a given cable diameter can be obtained from "Skyline Tension and Deflection Handbook" (see footnote 2).

The individual road specifications are made by describing the headspar and tailspar locations, the inner and outer yarding limits, and the range and elevation of enough profile points to characterize the ground under the skyline. The following convention was adopted for these data (fig. 1):

1. The yarder is always on the left at spar 1.
2. The terrain points are expressed as coordinates: the abscissa x denotes the horizontal distance measured from spar 1, or from some reference point to the left of spar 1, in feet; the ordinate y denotes the elevation of the point, in feet.

All of the above specifications are read in, and values are computed for internal variables in sections 1 and 2 of the main program.

In section 3, the geometric parameters, such as slope (S), span (L), the difference in elevation of the top of the spars (H), and the carriage's vertical (DY1) and horizontal (D1) position are determined from the input data.

In section 4 of the main program, each terrain point is checked to determine if it intersects the chord of the skyline and if it is within the yarding limits. These checks are performed before any skyline computations are made. If a terrain point is found to intersect the skyline chord, the computations for that skyline road are terminated, and a diagnostic statement is written. Skyline computations are performed only on terrain points that are within the yarding limit. The yarding limits can be located at the headspar, tailspar, or anywhere within the skyline span.

The line tensions and payload are computed in section 5 of the main program. Before these quantities can be determined, the geometric configuration of the skyline must be known because the analysis depends on the relative positions of spar 1, spar 2, and the carriage. The allowable tensions of the haulback line and the main line determine in which line the tension is critical, and the critical tension always occurs at the highest point reached by a given line.

If the yarder is at the upper end, as shown in figure 2, the carriage will always be below the top of spar 1, and the critical tension will occur at spar 1 (Type = 6), either in the main line or in the haulback line (Type = 8) depending on their relative sizes. If the yarder is at the lower end, one of the three possibilities depicted in figures 3, 4, or 5 can occur. The critical tension may occur in the haulback line at spar 2 (Type = 7), regardless of the position of the carriage,

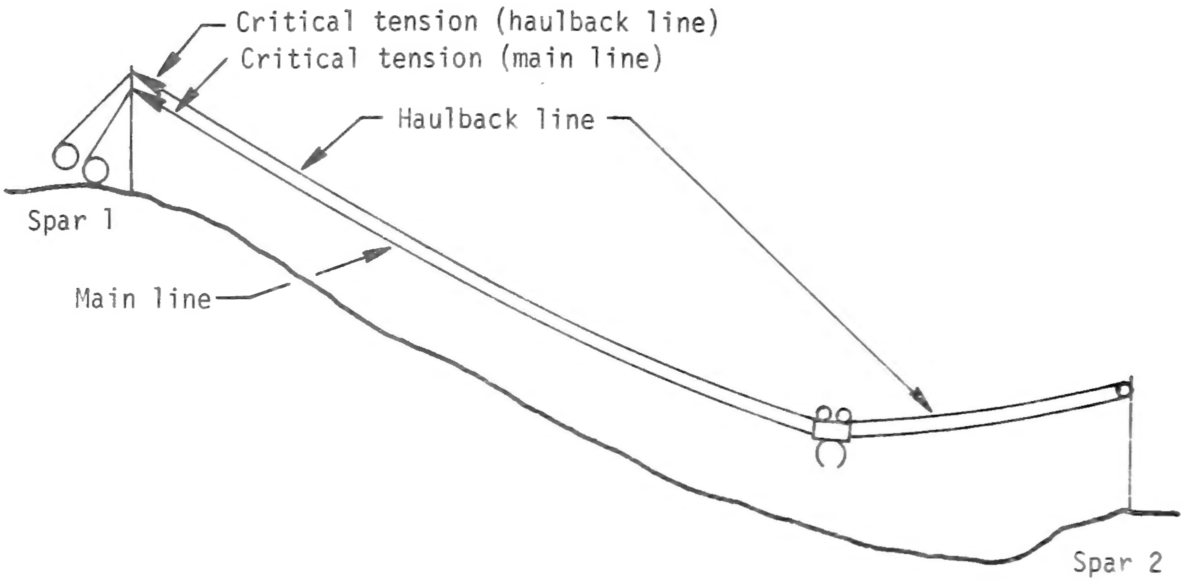


Figure 2.--*Yarder at upper end. Critical tension occurs at spar 1, either in the haulback line (Type = 6), or in the main line (Type = 8).*

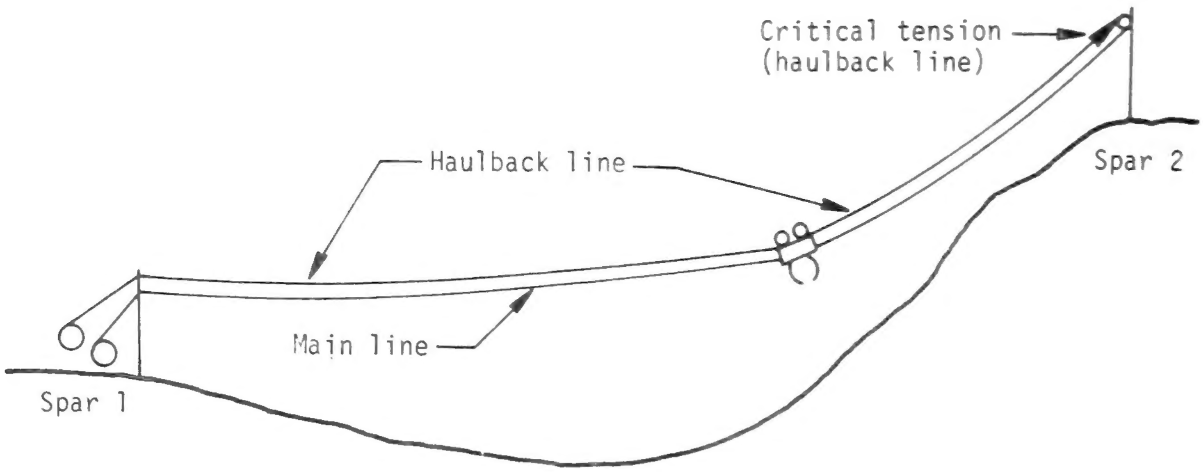


Figure 3.--*Yarder at lower end. Critical tension occurs at spar 2 in the haulback line (Type = 7).*

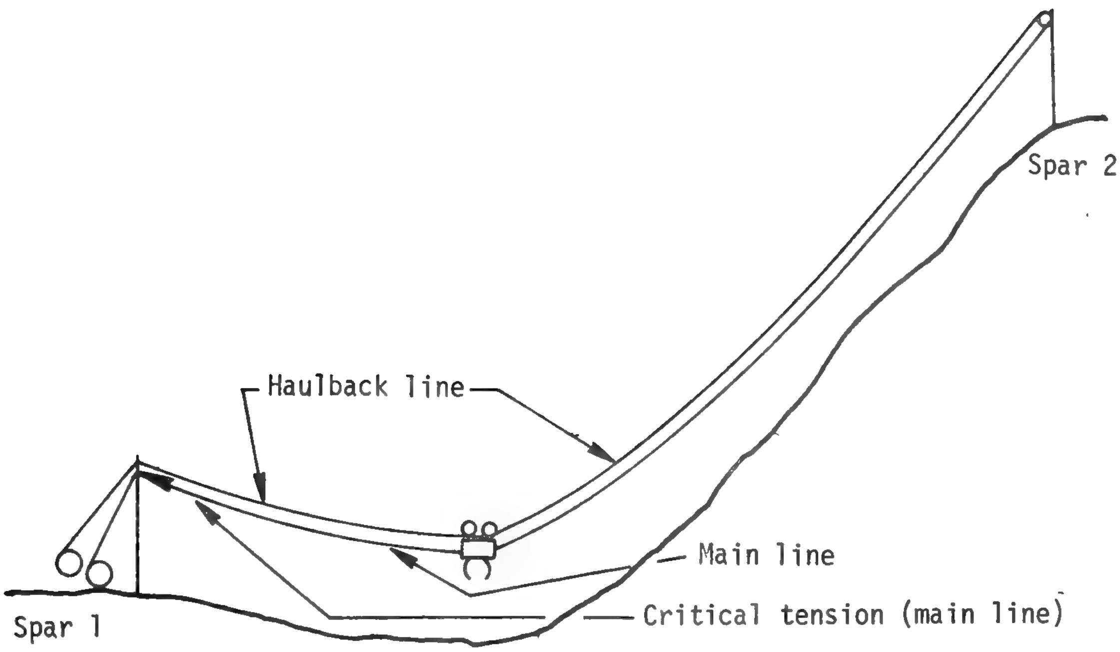


Figure 4.--*Yarder at lower end. Carriage below top of spar 1. Critical tension occurs at spar 1 in the main line (Type = 8).*

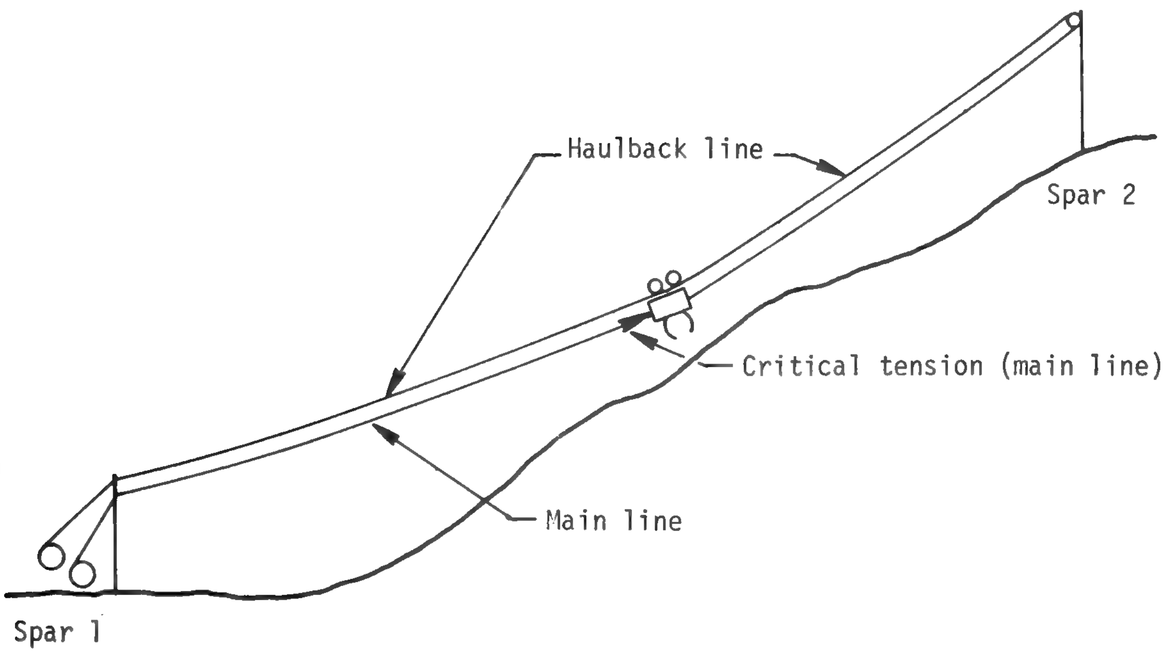


Figure 5.--*Yarder at lower end. Carriage above top of spar 1. Critical tension occurs at the carriage in the main line (Type = 9).*

as shown in figure 3. However, if the line sizes are such that the critical tension occurs in the main line, then the position of the carriage will determine its location. If the carriage is below the top of spar 1, the critical tension will occur at spar 1 (Type = 8), as shown in figure 4. If the carriage is positioned above the top of spar 1, the critical tension will occur at the carriage (Type = 9), as shown in figure 5.

After all the geometric parameters of the skyline road and the type are determined, control of the program is transferred to subroutine CONVG. This subroutine controls the iterations for solving the catenary equations (see footnote 4). The flow diagram illustrating the iterative procedure is shown in figure 6. The initial estimates for the catenary parameters m_1, m_2, and m_3 are found by solving for horizontal tensions in the force

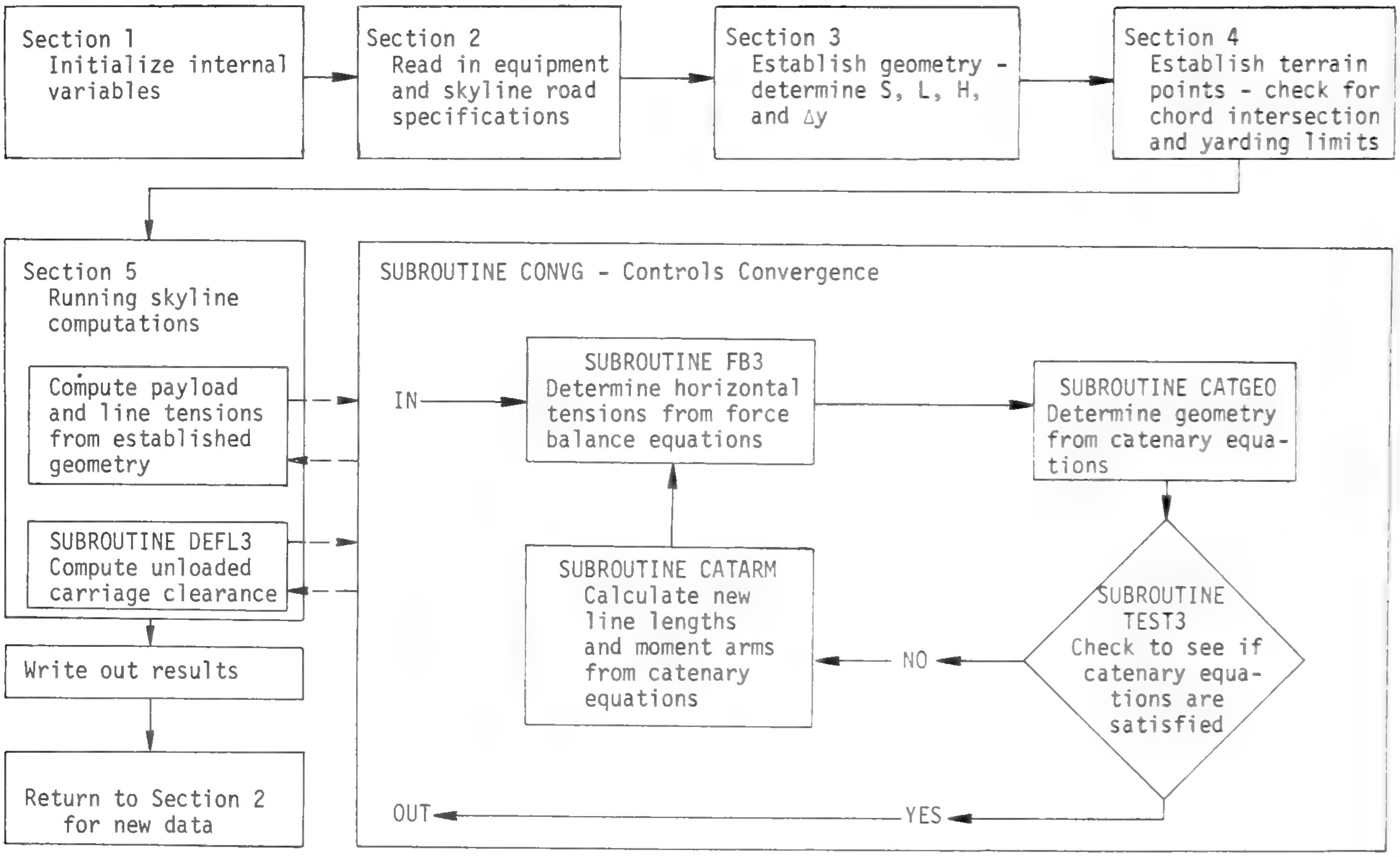


Figure 6.--*Macro flow diagram: Main program.*

balance equation. This is done in subroutine FB3. These initial estimates for the m's are then substituted in the catenary equations in subroutine CATGEO and tested for convergence in subroutine TEST3. The convergence equations are derived from the boundary conditions related to the type of skyline configuration. If the convergence functions are found to be within the tolerance set in TEST3, the control of the program is returned to the main program with the solution. If the convergence functions do not fall within the tolerance, new line lengths and moment arms are calculated in subroutine CATARM for use in FB3 to provide new estimates for the m's. The process is then repeated until a solution is found.

At this point, all conditions have been satisfied and the program execution could be terminated. However, one more step is taken. Determination of the unloaded carriage clearance at the given terrain point--that is, the distance from the ground to the carriage when the payload is equal to zero--is done in subroutine DEFL3. Determination of the unloaded carriage clearance involves Δy as an unknown. The iterative procedure of this routine employes the secant method. This method requires two initial estimates of the unknown Δy, plus a third estimate that is calculated from the secant formula. The first estimate of Δy is taken from the solution of the skyline problem along with the calculated payload. The second estimate of Δy is taken as a percentage of the first, and a new payload is calculated by entering subroutine CONVG. If the payload is found to be unequal to zero, the secant iteration loop is entered, a new Δy is determined from the secant formula, and subroutine CONVG is again entered to calculate a new payload. If this new payload is found to be unequal to zero, the secant formula is again used to calculate a new Δy, and the process is continued until the payload is found to equal zero.

INPUT

Input consists of the title, the equipment data, and the geometric data. Sample data forms are shown in figure 7. The title, which is contained on two cards, is read in an alphanumeric format. The first card contains one field of 70 columns for the project title and one field of 10 columns for the date. The second title card contains four fields: 10 columns for the region, 20 columns each for the Forest and District, and 30 columns for the location of the project.

The equipment data are also contained on two cards. The first card consists of eight fields of 10 columns, read by the Fortran 8F10. 1 format. Decimal points must be punched. The cable diameter, weight, and breaking strength for both the haulback line and main line, plus the safety factor and carriage weight, are contained on this card. The second equipment data card specifies the number of skyline roads that will accompany the set of title and equipment data, and the input type, read by the Fortran I25, I55 format.

The input type is coded zero if the geometric data are furnished by the designer in the form of x and y coordinates. If the geometric data are to be taken from aerial photographs, input type is coded 1. The procedure for submitting aerial photos is discussed in Research Note PNW-132.[5]

[5] Ward W. Carson, Donald D. Studier, and William M. Thomas. Digitizing topographic data for skyline design programs. USDA For. Serv. Res. Note PNW-132, 13 p., illus., 1970. Pac. Northwest For. & Range Exp. Stn., Portland, Oreg.

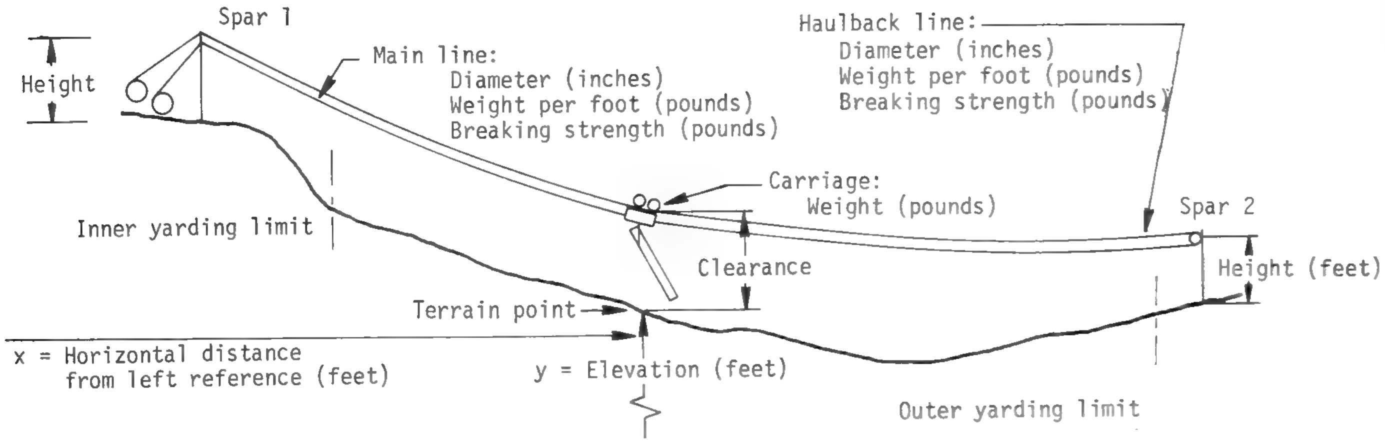
RUNNING SKYLINE

Project Title:			70	Date: 80
Region: 10	Forest: 30	District: 50	Location: 80	

Equipment Data

HAULBACK			MAIN LINE				
Cable Diameter	Cable Weight	Breaking Strength	Cable Diameter	Cable Weight	Breaking Strength	Safety Factor	Carriage Weight
10 xx.xx	20 xxx.xx	30 xxxxxx.xx	40 xx.xx	50 xx.xx	60 xxxxxx.xx	70 x.xx	80 xxxx.xx
.	.	.	.	.	.	.	.

Number of Skyline Roads 25	Format of Geometric Input Data 80



Geometric Data

Road Number	25

	LEFT SPAR			RIGHT SPAR			YARDING LIMIT	
Clear-ance	Height	x	y	Height	x	y	Inner	Outer
5 xxx	10 xxx	15 xxxx	20 xxxx	25 xxx	30 xxxx	35 xxxx	40 xxxxx	45 xxxxx

x = Horizontal distance from left reference point (feet)

y = Elevation (feet)

Clearance = Distance from the carriage to the ground (feet).

Terrain Data

x	y	x	y	x	y	x	y	x	y	x	y	x	y	x	y
5 xxxx	10 xxxx	15 xxxx	20 xxxx	25 xxxx	30 xxxx	35 xxxx	40 xxxx	45 xxxx	50 xxxx	55 xxxx	60 xxxx	65 xxxx	70 xxxx	75 xxxx	80 xxxx

The data to be provided here represent the geometric description of one skyline road. When more than one skyline road uses the same equipment and all roads are to be examined, other sets of these geometric data cards follow directly separated by one blank card. The title and equipment data cards need not be repeated for each skyline road.

Figure 7.--*Input data form.*

One set of geometric data describes an individual skyline road. The first geometric data card contains the skyline road number and is read by the Fortran I25 format. The second geometric card consists of nine fields of five columns each, read by the Fortran F5.0 format. This card contains the carriage clearance, height and location of the left and right spars, and the inner and outer yarding limits. The terrain data cards contain 16 fields of five columns each, read by the Fortran 16F5.0 format, and the values of distance (x) and elevation (y). There can be more than one terrain data card; the only requirement is that the last card be left blank. This blank card marks the end of the data for that skyline road, and the program will proceed to the next set of geometric data. When all sets of geometric data are read, the program will proceed to the next set of title and equipment data with its related geometric data.

OUTPUT

The output is illustrated in figure 8. The title and equipment data are printed in an orderly manner for easy reference. This is followed by the data for each skyline road, which consist of the input data and the solution to the problem.

The first section of the output consists of the title and equipment data that were furnished by the designer. It includes the project title, date, and location; the

```
                                   LEWIS RIVER SKYLINE SALE                              4/10/70

REGION 6/  FOREST     GIFFORD PINCHOT/  DISTRICT     LEWIS RIVER/  LOCATION     LEWIS RIVER

                              CABLE DIAMETER           LINE WEIGHT          BREAKING STRENGTH
HAULBACK                           .75                    1.04                   58800.00
MAIN LINE                         1.00                    1.85                  103400.00
SAFETY FACTOR = 3.00
CARRIAGE WEIGHT = 5000.00                   *************************
ROAD NUMBER = 1                             LOAD CLEARANCE =    20.0

                              SPAR HEIGHT              STATION                ELEVATION
 HEAD SPAR                        50.0                    0.0                   4826.0
 TAIL SPAR                        20.0                 1500.0                   4734.0

 YARDING LIMITS          0                  1500

SKYLINE SPAN = 1500.0          SLOPE =           -8.1 PERCENT

RUNNING SKYLINE OUTPUT AT ALL TERRAIN POINTS
```

HORIZONTAL DISTANCE	ELEVATION	PAYLOAD LBS.	DEFLECTION	-----------CABLE TENSIONS------------ HAULBACK YARDER	HAULBACK TAILSPAR	MAIN LINE YARDER	UNLOADED CARRIAGE CLEARANCE
179.0	4792.0	5563.9	3.30	19600.0	19473.2	21852.7	42.5
332.0	4771.0	2026.8	3.87	19600.0	19473.1	20728.4	33.5
510.0	4740.0	1836.0	4.97	19600.1	19473.1	20507.8	35.9
701.0	4702.0	3230.1	6.47	19600.1	19473.1	20436.6	51.2
891.0	4637.0	8574.0	9.77	19600.3	19473.2	20238.0	102.1
996.0	4570.0	15541.8	13.67	19600.4	19473.3	19203.2	165.0

Figure 8.--*Output format.*

diameter, weight, and breaking strength of the haulback and main lines; the safety factor; and the weight of the carriage. These data may be common to more than one skyline road.

The second section of the output applies to one particular road. The road number, load clearance, head spar and tail spar heights, station, elevation, and yarding limits are data that were furnished by the designer. The program calculates the skyline span, which is the horizontal distance between the head spar and tail spar, and the percent slope of the skyline chord.

The running skyline capability is calculated for each terrain point which falls within the yarding limits. If the terrain point is outside the yarding limit, a diagnostic statement is printed.

A discussion of the columns listed under "Running Skyline Output at all Terrain Points" (fig. 8) follows:

The first two columns, horizontal distance and elevation, describe the terrain point and are furnished by the designer or taken directly from the aerial photographs (see footnote 5).

The calculated payloads, listed in the third column, are the payloads that the skyline will support when the carriage is positioned at the required clearance above the terrain points.

Deflection is the vertical distance from the chord to the skyline, expressed as a percent of the span length. This program calculates the deflection in percent, at the terrain point in question, when the skyline is supporting the calculated payload.

The next three columns list the cable tensions that occur at the yarder and tailspar in both the main and haulback lines when they are supporting the calculated payload.

In the last column, the unloaded carriage clearance is listed for each terrain point. This is the distance between the carriage and the ground when the carriage is unloaded and the working tension is maintained in the lines.

NOMENCLATURE FOR THE RUNNING SKYLINE COMPUTER PROGRAM

ARG1	Intermediate quantity used in the computation of horizontal tension, pounds squared.
ARG2	Intermediate quantity used in the computation of horizontal tension, pounds squared.
AX, AY, AZ	Coordinates of terrain points, determined from aerial photographs.
BX, BY, BZ	Coordinates of locations of the head- and tailspars and the inner and outer yarding limits, determined from aerial photographs.
C1	Required carriage clearance, feet.
CACL	Distance between skyline chord and specified carriage location, feet.
CAEL	Elevation of carriage when it is at the specified clearance above the terrain point, feet.
CDSKY	Cable diameter of haulback line, inches.
CHEL	Elevation of skyline chord, feet.
DATE	Input title storage.
DEFL	Deflection at given terrain point.
D1	Horizontal location of terrain point from spar 1, feet.
DF	Deflection, percent.
DUM1	Intermediate quantity used in the computation of horizontal tension, pounds squared.
DUM2	Intermediate quantity used in the computation of horizontal tension, pounds squared.
DY1	Vertical distance from spar 1 to carriage when it is located at the specified clearance distance above the terrain point, feet.
E1	Moment arm of the weight of line segment 1 from the carriage, feet.
E2	Moment arm of the weight of line segment 2 from the carriage, feet.
ES	Moment arm of the weight of the main line from the carriage, feet.

ES1	Elevation of spar 1, feet.
ES2	Elevation of spar 2, feet.
FM ( )	Convergence functions for the various ITYPEs.
H	Difference in elevation between top of spar 1 and top of spar 2, feet.
H1	Horizontal component of tension in line segment 1 of the haulback line, pounds.
H2	Horizontal component of tension in line segment 2 of the haulback line, pounds.
HS	Horizontal component of tension in the main line, pounds.
HS1	Height of spar 1, feet.
HS2	Height of spar 2, feet.
I	Indicator that functions have converged.
IBO	Indicator, if equal to 1, causes intermediate output to be printed.
IDUM	Indicator for ITYPE.
ILI	Indicator noting the first terrain point inside the inner yarding limit.
IL2	Indicator noting the last terrain point inside the outer yarding limit.
ILT	Indicator used in routine accounting for yarding limits.
IND	Indicator used to eliminate some calculations after the first iteration.
INTYP	Indicator noting if the input data is two-dimensional or three-dimensional.
L	Horizontal distance between spar 1 and spar 2, feet.
SS2	Horizontal location of spar 2, feet.
ST1	Ratio of DY and D.
ST2	Ratio of E1 and D.
ST3	Ratio of ES and DS.
ST4	Ratio of (DY-H) and (L-D).

ST5	Ratio of E2 and (L-D).
ST6	Ratio of SYS and DS.
TA	Allowable tension in haulback, pounds.
TAS	Allowable tension in main line, pounds.
TBSKY	Breaking strength of haulback line, pounds.
TBSNUB	Breaking strength of main line, pounds.
TITLE1	Input title storage.
TITLE2	Input title storage.
TOL	Tolerance allowed in the convergence function.
V1	Vertical component of tension in line segment 1 of the skyline, pounds.
V2	Vertical component of tension in line segment 2 of the skyline, pounds.
VS	Vertical component of tension in the main line, pounds.
WC	Weight of the carriage, pounds.
WL	Payload or weight of logs, pounds.
WO1	Weight of segment 1 of haulback, pounds/foot.
WO2	Weight of segment 2 of haulback, pounds/foot.
WOS	Weight of main line, pounds/foot.
X	Horizontal location of terrain point, feet.
X1MO1	Geometric quantity used in the solution of catenary equations.
XAMO1	Geometric quantity used in the solution of catenary equations.
X2MO2	Geometric quantity used in the solution of catenary equations.
XAMO2	Geometric quantity used in the solution of catenary equations.
X1MOS	Geometric quantity used in the solution of catenary equations.
XAMOS	Geometric quantity used in the solution of catenary equations.
Y	Elevation of terrain point, feet.
YL1	Horizontal location of inner yarding limit, feet.
YL2	Horizontal location of outer yarding limit, feet.

RUNNING SKYLINE PROGRAM

```
 C      PROGRAM  RSKY
 C      ITYPE=6 H.GE.0
C     ITYPE=7 H.LT.0
C     ITYPE=8 DY.GE.0
C     ITYPE=9 DY.LT.0
C  MACHINE DATA
C   TBSKY =BREAKING STRENGTH OF HAULBACK,POUNDS
C   TBSNUB=BREAKING STRENGTH OF MAIN LINE,POUNDS
C   CDSKY =DIAMETER OF HAULBACK,INCHES
C   CDSNUB=DIAMETER OF MAIN LINE,INCHES
C   WO1   =WEIGHT OF HAULBACK,POUNDS/FOOT
C   WOS   =WEIGHT OF MAIN LINE,POUNDS/FOOT
C   SF    =DESIGN SAFETY FACTOR
C   WC    =WEIGHT OF CARRIAGE,POUNDS
C   HS1   =HEIGHT OF SPAR 1,FEET
C   HS2   =HEIGHT OF SPAR 2,FEET
C  MISCELLANEOUS DATA
C   YL1=INNER YARDING LIMIT,FEET
C   YL2=OUTER YARDING LIMIT,FEET
C   ES1=ELEVATION OF SPAR 1,FEET
C   ES2=ELEVATION OF SPAR 2,FEET
C   SS1=STATION OF SPAR 1,FEET
C   SS2=STATION OF SPAR 2,FEET
C   IRN=SKYLINE ROAD NUMBER
C
 C  GEOMETRY  -  ALL IN FEET
C   S  =SLOPE
C   L  =SPAN
C   DI =STATION
C   DYI=DISPLACEMENT FROM TOP OF SPAR 1 TO CARRIAGE
C   H  =DISPLACEMENT FROM TOP OF SPAR 1 TO TOP OF SPAR 2
C   Y  =TERRAIN POINT ELEVATION
C   X  =TERRAIN POINT STATION
C   C1 =REQUIRED CARRIAGE CLEARANCE
      DIMENSION X(50),Y(50),DI(50),DYI(50),CACL(50),WP1(50),DF(50),CL(50

     1)
      DIMENSION AX(50),AY(50),AZ(50),IA(50)
      DIMENSION BX(50),BY(50),BZ(50),IB(50)
      DIMENSION TITLE1(16),TITLE2(16),DATE(2)
      DIMENSION T1(50),T2(50),TS(50)
      REAL L,L1,L2,LS,MO1,MO2,MOS
      COMMON IPRINT,ITER
      COMMON TA,TAS,WO1,WO2,WOS,L,S,DEFL,D,DY,DS,H,DYS,WC,C1,HS1,HS2
      COMMON L1,L2,LS,E1,E2,ES
      COMMON MO1,MO2,MOS,XAMO1,X1MO1,XAMO2,X2MO2,XAMOS,X1MOS
      COMMON WL,SG
      INTEGER OT
C  SECTION 1 ***** INITIALIZE
      IN=60
      OT=61
      NP=62
```

```
      IPRINT=1
      IPRINT=0
C
C  SECTION 2 *****INPUT - OUTPUT
 1    CONTINUE
      READ (IN,1005) (TITLE1(I),I=1,14),DATE(1),DATE(2)
      GO TO (5000,2)  EOFCKF(IN)
    2 READ (IN,1005) (TITLE2(I),I=1,16)
      ISET=0
      WRITE (OT,2001) (TITLE1(I),I=1,12),DATE(1),DATE(2)
      WRITE (OT,2002) (TITLE2(I),I=1,16)
C
C  EQUIPMENT SPECIFICATIONS
      READ (IN,1000) CDSKY,WO1,TBSKY,CDSNUB,WOS,TBSNUB,SF,WC
      SF=3.0
      WRITE (OT,2004) CDSKY,WO1,TBSKY,CDSNUB,WOS,TBSNUB,SF,WC
      TA=TBSKY/SF
      TAS=TBSNUB/SF
      WO2=WO1
      READ(IN,1001) NOR,SCL,INTYP
C
C  SKYLINE ROAD SPECIFICATIONS
 3    IF (INTYP.EQ.0) GO TO 9
C
C  AUTO-TROL DATA
      IF(SCL.EQ.0.0) SCL=1.0
      READ (IN,1004) IRN,C1,HS1,HS2,ICARD,(BX(I),BY(I),BZ(I),IB(I),
     1I=1,4)
      N1=1
 4    N2=N1+3
      READ (IN,1006) (AX(I),AY(I),AZ(I),IA(I),I=N1,N2)
      N1=N1+4
      IF (IA(N2).EQ.2) GO TO 4
 5    IF (IA(N2).EQ.3) GO TO 6
      N2=N2-1
      GO TO 5
 6    N=N2
      SS1=SQRT((BX(1)-AX(1))**2+(BY(1)-AY(1))**2)*SCL
      ES1=BZ(1)
      YL1=SQRT((BX(2)-AX(1))**2+(BY(2)-AY(1))**2)*SCL
      YL2=SQRT((BX(3)-AX(1))**2+(BY(3)-AY(1))**2)*SCL
      SS2=SQRT((BX(4)-AX(1))**2+(BY(4)-AY(1))**2)*SCL
      ES2=BZ(4)
      WRITE (OT,2005) IRN,C1,HS1,SS1,ES1,HS2,SS2,ES2,YL1,YL2
      DO 7 I=1,N
      X(I)=SQRT((AX(I)-AX(1))**2+(AY(I)-AY(1))**2)*SCL
      Y(I)=AZ(I)
 7    CONTINUE
      WRITE(NP,1001) IRN
      WRITE (NP,1003) C1,HS1,SS1,ES1,HS2,SS2,ES2,YL1,YL2
      WRITE(NP,1002)(X(I),Y(I),I=1,N)
      WRITE(NP,7000)
      GO TO 30
C
C  TERRAIN DATA-NOTE,ONE SET OF BLANKS REQUIRED AFTER LAST SET OF TERRAIN

 9    READ (IN,1001) IRN
      GO TO (1,91) EOFCKF(IN)
 91   READ (IN,1003) C1,HS1,SS1,ES1,HS2,SS2,ES2,YL1,YL2
```

```
      WRITE (OT,2005) IRN,C1,HS1,SS1,ES1,HS2,SS2,ES2,YL1,YL2
      N1=1
      N2=8
 10   READ (IN,1002) (X(I),Y(I),I=N1,N2)
      DO 11 I=N1,N2
      IF (Y(I).EQ.0.0) GO TO 12
 11   N=I
      N1=N1+8
      N2=N2+8
      GO TO 10
   12 IF(N.NE.(N1-1))  READ (IN,1001)
C
 C  SECTION 3 ***** ESTABLISH GEOMETRY
 30   L=SS2-SS1
      S=((ES1+HS1)-(ES2+HS2))/L
      SP=-100.0*S
      H=S*L
      WRITE (OT,2006) L,SP
      DO 32 I=1,N
      DI(I)=X(I)-SS1
      DYI(I)=(ES1+HS1)-Y(I)-C1
 32   CONTINUE
C  CHECK FOR YARDING LIMITS OUTSIDE OF THE ANCHOR POINTS
      IF(YL1.LT.SS1.OR.YL2.GT.SS2) WRITE(OT,2019)
      IF(YL1.LT.SS1.OR.YL2.GT.SS2) GO TO 100
C
 C  SECTION 4 **** ESTABLISH TERRAIN POINTS
C
       IL1=1
      IL2=N
      ISET=0
C     CHECK YARDING LIMITS
      DO 40 I=1,N
      IF (X(I).LE.YL1) GO TO 45
      IF (X(I).GE.YL2) GO TO 46
      IF(X(I).LT.SS1.OR.X(I).GT.SS2) GO TO 421
      GO TO 47
 45   CONTINUE
      IL1=I+1
      WRITE (OT,2010) X(I),Y(I)
      GO TO 40
 46   CONTINUE
      ILT=I-1
      IL2=MIN0(IL2,ILT)
      WRITE (OT,2010) X(I),Y(I)
      GO TO 40
C
  47   CONTINUE
C   CHECK TERRAIN FOR CHORD INTERSECTION
      CHEL=ES1+HS1-DI(I)*S
      CAEL=Y(I)+C1
      CACL(I)=CHEL-CAEL
      DF(I)=CACL(I)/L*100.0
      DEFL=CACL(I)/L
      IF (CACL(I).LE.0.0) ISET=1
       GO TO 40
 421  WRITE(OT,2017) X(I),Y(I)
      GO TO 100
   40 CONTINUE
```

```
      IF (ISET.EQ.1) GO TO 61
C
 C  SECTION 5 **** RUNNING SKYLINE COMPUTATIONS
C
       DO 60 I=IL1,IL2
      D=DI(I)
      DS=D
      DY=DYI(I)
      DYS=DY
C           DETERMINE ITYPE
      IF (H.GE.0.0) GO TO 50
      ITYPE=7
      GO TO 51
 50   ITYPE=6
 51   CALL  CONVG(ITYPE)
C
 C          CHECK DY
      IF(DY.GE.0.0) GO TO 52
      TDIFF=TAS-WOS*MOS*COSH(XAMOS)
      IF (TDIFF.GE.0.0) GO TO 54
      ITYPE=9
      GO TO 53
 52   TDIFF=TAS-WOS*MOS*COSH(X1MOS)
      IF (TDIFF.GE.0.0) GO TO 54
      ITYPE=8
 53   CALL  CONVG(ITYPE)
 54   WP1(I)=WL
      T1(I)=MO1*WO1*COSH(X1MO1)
      T2(I)=MO2*WO2*COSH(X2MO2)
      TS(I)=MOS*WOS*COSH(X1MOS)
      CALL DEFL3(ITYPE)
      CL(I)=HS1+ES1-Y(I)-(D*S+DEFL*L)
   60 CONTINUE
   61 CONTINUE
      IF (ISET.EQ.1) WRITE (OT,3000)(X(I),Y(I),CACL(I),I=IL1,IL2)
      IF (ISET.EQ.1) GO TO 100
 99   CONTINUE
      WRITE (OT,2007)
      WRITE (OT,3001) (X(I),Y(I),WP1(I),DF(I),T1(I),T2(I),TS(I),CL(I),I=

     1IL1,IL2)
 100  CONTINUE
      GO TO 3
 5000 RETURN
 1000 FORMAT(8F10.0)
 1001 FORMAT(20X,I5,5X,F10.0,35X,I5)
 1002 FORMAT (16F5.0)
 1003 FORMAT (16F5.0)
 1004 FORMAT (I5,3F3.0,I1,4(3F5.0,I1))
 1005 FORMAT (16A5)
 1006 FORMAT (15X,4(3F5.0,I1))
 2001 FORMAT (1H1,20X,12A5,20X,2A5)
 2002 FORMAT (8H REGION ,2A5,9H/ FOREST ,4A5,11H/ DISTRICT ,4A5,
     1                                          12H/  LOCATION ,6A5/)

  2004 FORMAT (1H ,20X,15H CABLE DIAMETER,5X,12H LINE WEIGHT,8X,18H BREA

        1ING STRENGTH/9H HAULBACK,5X,3F20.2/9H MAINLINE,5X,3F20.2/15H S
```

```
    2Y FACTOR=,F5.2/17H CARRIAGE WEIGHT=,F8.1//)
2005 FORMAT (1X,//,40X,20H********************,//,
    3         13H ROAD NUMBER=,I5,20X,16H LOAD CLEARANCE=,F5.1/23X,12H S
    1PAR HEIGHT,10X,8H STATION,9X,10H ELEVATION/12H   HEAD SPAR,3F20.1/
    212H   TAIL SPAR,3F20.1//3X,14HYARDING LIMITS,2F10.)
2006 FORMAT (15H0 SKYLINE SPAN=,F7.1,5X,7H SLOPE=,F6.1,8H PERCENT//)
2007 FORMAT (45H RUNNING SKYLINE OUTPUT AT ALL TERRAIN POINTS/,
    171X,42H------------- CABLE TENSIONS -------------,7X,8HUNLOADED/
     2,3X,10HHORIZONTAL,25X,7HPAYLOAD,26X,8HHAULBACK,9X,8HHAULBACK,9X,
    38HMAINLINE,7X,8HCARRIAGE/,4X,8HDISTANCE,8X,9HELEVATION,11X,4HLBS.,
    46X,10HDEFLECTION,12X,6HYARDER,10X,8HTAILSPAR,10X,6HYARDER,8X,
    59HCLEARANCE)
2010 FORMAT (2F9.1,34H THIS POINT OUTSIDE YARDING LIMITS)
2012 FORMAT (2F9.1,2(2F9.1,F9.4),F9.1)
2013 FORMAT (2F9.1,2(2F9.1,F9.4),F9.1,18H HAULBACK REQUIRED)
2017 FORMAT(2F9.1,75H THIS POINT OCCURS OUTSIDE OF THE ANCHOR POINTS.
     $CHECK THE YARDING LIMITS.
2019 FORMAT (/18X,61H THE YARDING LIMITS MUST NOT BE OUTSIDE OF THE ANC
    $HOR POINTS   /)
3000 FORMAT(79H0CARRIAGE WILL NOT CLEAR THE GROUND AT TERRAIN POINTS WI
    $TH NEGATIVE CLEARANCE   /30H0  STATION ELEVATION CLEARANCE //
    $(3F10.0)/)
3001 FORMAT (F11.1,2(5X,F12.1),5X,F8.2,8XF12.1,3(5X,F12.1)/)
7000 FORMAT(1H )
     END

      SUBROUTINE  CONVG(ITYPE)
C  ROUTINE TO CONTROL ITERATIONS BETWEEN FORCE BALANCE AND CATENARY
C
      REAL L,L1,L2,LS,MO1,MO2,MOS
     COMMON IPRINT,ITER
     COMMON TA,TAS,WO1,WO2,WOS,L,S,DEFL,D,DY,DS,H,DYS,WC,C1,HS1,HS2
     COMMON L1,L2,LS,E1,E2,ES
     COMMON MO1,MO2,MOS,XAMO1,X1MO1,XAMO2,X2MO2,XAMOS,X1MOS
     COMMON WL,SG
C
 C  START ITERATION LOOP
C
      IND=1
     ITER=0
10   ITER=ITER+1
C
 C  GET FORCE BALANCE ESTIMATES OF HORIZONTAL TENSIONS
     CALL FB3(IND,ITYPE)
C
 C  GENERATE CATENARY GEOMETRY
     CALL CATGEO
C
 C  CHECK CONVERGENCE
```

```
      CALL TEST3(ITYPE,I)
      IND=0
C
       IF (ITER.EQ.10) RETURN
      IF (I.EQ.1) GO TO 20
      RETURN
 20   CALL CATARM
      GO TO 10
      END

      SUBROUTINE FB3(IND,ITYPE)
C
       REAL L,L1,L2,LS,MO1,MO2,MOS
      COMMON IPRINT,ITER
      COMMON TA,TAS,WO1,WO2,WOS,L,S,DEFL,D,DY,DS,H,DYS,WC,C1,HS1,HS2
      COMMON L1,L2,LS,E1,E2,ES
      COMMON MO1,MO2,MOS,XAMO1,X1MO1,XAMO2,X2MO2,XAMOS,X1MOS
      COMMON WL,SG
      INTEGER OT
      OT=61
C  COMPUTE NECESSARY COEFFICIENTS
C  INITIALIZE IF NECESSARY
      IF (IND.NE.1) GO TO 10
C  INITIALIZE LINE LENGTHS AND MOMENT ARMS
      L1=SQRT(D**2+DY**2)
      L2=SQRT((DY-H)**2+(L-D)**2)
      LS=SQRT(DS**2+DYS**2)
      E1=D/2.0
      E2=(L-D)/2.0
      ES=DS/2.0
 10   ST1=DY/D
      ST2=E1/D
      ST3=ES/DS
      ST4=(DY-H)/(L-D)
      ST5=E2/(L-D)
      ST6=DYS/DS
C  ESTABLISH LINE WEIGHTS
      R1=L1*WO1
      R2=L2*WO2
      RS=LS*WOS
      IDUM = ITYPE - 5
      GO TO (15,20,25,30),IDUM
C
 C   FOR H.GE.0 - COMPUTE PAYLOAD TYPE=6
C
  15   CONTINUE
C  COMPUTE MO1
      ARG1=(R1*ST1*ST2)**2-(1.0+ST1**2)*((R1*ST2)**2-TA**2)
      IF (ARG1.LT.0.0) GO TO 90
      H1=(-R1*ST1*ST2+SQRT(ARG1))/(1.0+ST1**2)
      MO1=H1/WO1
C
 C  COMPUTE MO2
      V1=R1*ST2+H1*ST1
      DUM1=(V1-R1)**2+H1**2
      ARG2=(R2*ST4*(ST5-1.0))**2-(1.0+ST4**2)*((R2*(ST5-1.0))**2-DUM1)

       IF (ARG2.LT.0.0) GO TO 90
      H2=(-R2*ST4*(ST5-1.0)+SQRT(ARG2))/(1.0+ST4**2)
```

```
      MO2=H2/WO2
C
 C  COMPUTE MOS
      HS=2*H2-H1
      MOS=HS/WOS
C
 C  COMPUTE WEIGHT OF LOG
      V2=R2*ST5+H2*ST4
      VS=RS*ST3+HS*ST6
      WL=V1-R1+VS-RS+2.0*(V2-R2)-WC
      IF (IPRINT.EQ.1) GO TO 91
      RETURN
C
 C      FOR H.LT.0 - COMPUTE PAYLOAD TYPE=7
C
 20   CONTINUE
C
 C  COMPUTE MO2
      ARG1=(R2*ST4*ST5)**2-(1.0+ST4**2)*((R2*ST5)**2-TA**2)
      IF (ARG1.LT.0.0) GO TO 90
      H2=(-R2*ST5*ST4+SQRT(ARG1))/(ST4**2+1.0)
      MO2=H2/WO2
C       COMPUTE MO1
      V2=R2*ST5+H2*ST4
      DUM1= (V2-R2)**2+H2**2
      ARG2=(R1*ST1*(ST2-1.0))**2-(1.0+ST1**2)*((R1*(ST2-1.0))**2-DUM1)

       IF (ARG2.LT.0.0) GO TO 90
      H1=(-R1*ST1*(ST2-1.0)+SQRT(ARG2))/(1.0+ST1**2)
      MO1=H1/WO1
C  COMPUTE MOS
      HS=2.0*H2-H1
      MOS=HS/WOS
C  COMPUTE WEIGHT OF LOG
      V1=R1*ST2+H1*ST1
      VS=RS*ST3+HS*ST6
      WL=V1-R1+VS-RS+2.0*(V2-R2)-WC
      IF (IPRINT.EQ.1) GO TO 91
      RETURN
C
 C    FOR DY.GE.0 - TAS AT SPAR 1 TYPE=8
C
 25   CONTINUE
C  COMPUTE MOS
      ARG1=(RS*ST3*ST6)**2-(1.0+ST6**2)*((RS*ST3)**2-TAS**2)
      IF (ARG1.LT.0.0) GO TO 90
      HS=(-RS*ST3*ST6+SQRT(ARG1))/(1.0+ST6**2)
      MOS=HS/WOS
C
 C  COMPUTE MO2
      DUM1=2.0*R1*ST1*(ST2-1.0)-R2*(ST5-1.0)*ST4-2.0*HS*(1.0+ST1**2)
      DUM2=(R1*(ST2-1.0))**2+HS**2+(HS*ST1)**2-2.0*R1*(ST2-1.0)*ST1*HS

     1                                                        -(R2*(ST5-1.0))**2

      ARG2=DUM1**2-(4.0*(ST1**2+1.0)-(ST4**2+1.0))*DUM2
      IF (ARG2.LT.0.0) GO TO 90
      H2=(-DUM1+SQRT(ARG2))/(4.0*(ST1**2+1.0)-(ST4**2+1.0))
      MO2=H2/WO2
```

```
C
 C    COMPUTE MO1
      H1=2.0*H2-HS
      MO1=H1/WO1
C
 C   COMPUTE WEIGHT OF LOG
      V1=R1*ST2+H1*ST1
      V2=R2*ST5+H2*ST4
      VS=RS*ST3+HS*ST6
      WL=V1-R1+VS-RS+2.0*(V2-R2)-WC
      IF (IPRINT.EQ.1) GO TO 91
      RETURN
C
 C   FOR DY.LT.0 - TAS AT CARRIAGE TYPE=9
C
  30   CONTINUE
C  COMPUTE MOS
      ARG1=(RS*ST6*(ST3-1.0))**2-(1.0+ST6**2)*(RS**2*(ST3-1.0)**2
     1                                                     -TAS**2)
      IF (ARG1.LT.0.0) GO TO 90
      HS=(-RS*ST6*(ST3-1.0)+SQRT(ARG1))/(1.0+ST6**2)
      MOS=HS/WOS
C    COMPUTE MO2
      DUM1=2.0*R1*ST1*(ST2-1.0)-R2*(ST5-1.0)*ST4-2.0*HS*(1.0+ST1**2)
      DUM2=(R1*(ST2-1.0))**2+HS**2+(HS*ST1)**2-2.0*R1*(ST2-1.0)*ST1*HS

      1                                                   -(R2*(ST5-1.0))**

          ARG2=DUM1**2-(4.0*(ST1**2+1.0)-(ST4**2+1.0))*DUM2
      IF (ARG2.LT.0.0) GO TO 90
      H2=(-DUM1+SQRT(ARG2))/(4.0*(ST1**2+1.0)-(ST4**2+1.0))
      MO2=H2/WO2
C    COMPUTE MO1
      H1=2.0*H2-HS
      MO1=H1/WO1
C    COMPUTE WEIGHT OF LOG
      V1=R1*ST2+H1*ST1
      V2=R2*ST5+H2*ST4
      VS=RS*ST3+HS*ST6
      WL=V1-R1+VS-RS+2.0*(V2-R2)-WC
      IF (IPRINT.EQ.1) GO TO 91
      RETURN
 91   WRITE (OT,2001) ITYPE,ARG1,ARG2,ST1,ST2,ST3,ST4,ST5,ST6,L1,L2,LS,

     1       E1,E2,ES,R1,R2,RS,DUM1,DUM2,H1,H2,HS,V1,V2,VS,WC,WL
      RETURN
C
  90   WRITE (OT,2000) ITYPE,ARG1,ARG2,ST1,ST2,ST3,ST4,ST5,ST6,L1,L2,LS,

     1       E1,E2,ES,R1,R2,RS,DUM1,DUM2,H1,H2,HS,V1,V2,VS,WC
      RETURN
 2000 FORMAT (24H NEG ARG IN FB3    TYPE=,I2,6E15.7/3(1X,8E15.7/))
 2001 FORMAT (6H TYPE=,I2,6E15.7/3(1X,8E15.7/))
      END

      SUBROUTINE TEST3(ITYPE,I)
C
 C   PROGRAM TO CHECK CONVERGENCE
C
```

```
      REAL L,L1,L2,LS,MO1,MO2,MOS
      COMMON IPRINT,ITER
      COMMON TA,TAS,WO1,WO2,WOS,L,S,DEFL,D,DY,DS,H,DYS,WC,C1,HS1,HS2
      COMMON L1,L2,LS,E1,E2,ES
      COMMON MO1,MO2,MOS,XAMO1,X1MO1,XAMO2,X2MO2,XAMOS,X1MOS
      COMMON WL,SG
      INTEGER OT
      OT=61
C
       WL=MOS*WOS*SINH(XAMOS)+MO1*WO1*SINH(XAMO1)+2.0*MO2*WO2*SINH(XAMO2
     1                                                                )-WC
      TOL=0.5
      I=2
      IDUM=ITYPE-5
      GO TO (10,20,30,40),IDUM
C
 C  PORTION TREATING H.GE.0 - ITYPE = 6
C
  10   CONTINUE
C  TEST FM7 AND FM8
      FM7=TA-WO1*DY-WO1*MO1*COSH(XAMO1)
      FM8=TA-WO1*DY-WO2*MO2*COSH(XAMO2)
      IF (ABS(FM7)-TOL) 11,11,12
 11   IF (ABS(FM8)-TOL) 13,13,12
 12   I=1
C
  13   IF (ITER.EQ.10) GO TO 50
      IF (IPRINT.EQ.1) GO TO 51
      RETURN
C
 C  PORTION TREATING H.LT.0 - ITYPE 7
C
  20   CONTINUE
      FM9=TA-WO1*MO1*COSH(XAMO1)-WO2*(DY-H)
      FM10=TA-WO2*(DY-H)-WO2*MO2*COSH(XAMO2)
C   TEST FM9 AND FM10
      IF (ABS(FM9)-TOL) 21,21,22
 21   IF (ABS(FM10)-TOL) 23,23,22
 22   I=1
C
  23   IF (ITER.EQ.10) GO TO 60
      IF (IPRINT.EQ.1) GO TO 61
      RETURN
C
 C   PROTION TREATING DY.GE.0 - ITYPE 8
 30   CONTINUE
C  TEST FM11 AND FM12
      FM11=MO1*WO1*COSH(XAMO1)-MO2*WO2*COSH(XAMO2)
      FM12=TAS-WOS*DY-WOS*MOS*COSH(XAMOS)
      IF (ABS(FM11)-TOL) 31,31,32
C
  31   IF (ABS(FM12)-TOL) 33,33,32
 32   I=1
 33   IF (ITER.EQ.10) GO TO 70
      IF (IPRINT.EQ.1) GO TO 71
      RETURN
C
```

```
C   PORTION TREATING DY.LT.0 - ITYPE = 9
 40   CONTINUE
C  TEST FM11 AND FM13
      FM11=M01*W01*COSH(XAM01)-M02*W02*COSH(XAM02)
      FM13=TAS-MOS*WOS*COSH(XAMOS)
      IF (ABS(FM11)-TOL) 41,41,42
 41   IF (ABS(FM13)-TOL) 43,43,42
 42   I=1
C
  43   IF (ITER.EQ.10) GO TO 80
      IF (IPRINT.EQ.1) GO TO 81
      RETURN
C
  50   WRITE (OT,2000)
 51   CONTINUE
      WRITE (OT,3003) ITER,FM7,FM8,L1,L2,LS,E1,E2,ES
      WRITE (OT,3002) M01,M02,MOS,XAM01,X1M01,XAM02,X2M02,XAMOS,X1MOS

      GO TO 90
 60   WRITE (OT,2001)
 61   CONTINUE
      WRITE (OT,3004) ITER,FM9,FM10,L1,L2,LS,E1,E2,ES
      WRITE (OT,3002) M01,M02,MOS,XAM01,X1M01,XAM02,X2M02,XAMOS,X1MOS

      GO TO 90
 70   WRITE (OT,2002)
 71   CONTINUE
      WRITE (OT,3005) ITER,FM11,FM12,L1,L2,LS,E1,E2,ES
      WRITE (OT,3002) M01,M02,MOS,XAM01,X1M01,XAM02,X2M02,XAMOS,X1MOS

      GO TO 90
 80   WRITE (OT,2003)
 81   CONTINUE
      WRITE (OT,3006) ITER,FM11,FM13,L1,L2,LS,E1,E2,ES
      WRITE (OT,3002) M01,M02,MOS,XAM01,X1M01,XAM02,X2M02,XAMOS,X1MOS

 90   CONTINUE
C
 C  COMPUTE CATENARY TENSIONS
      T1=W01*M01*COSH(X1M01)
      T2=W02*M02*COSH(X2M02)
      TS=WOS*MOS*COSH(X1MOS)
      V1=W01*M01*SINH(XAM01)
      V2=W02*M02*SINH(XAM02)
      VS=WOS*MOS*SINH(XAMOS)
      H1=W01*M01
      H2=W02*M02
      HS=WOS*MOS
      WRITE (OT,3001) T1,T2,TS,V1,V2,VS,H1,H2,HS
      RETURN
 2000 FORMAT (35H TOO MANY ITERATIONS ON H.GE.0 CASE)
 2001 FORMAT (35H TOO MANY ITERATIONS ON H.LT.0 CASE)
 2002 FORMAT (36H TOO MANY ITERATIONS ON DY.GE.0 CASE)
 3001 FORMAT (29H (T1,T2,TS,V1,V2,VS,H1,H2,HS),9F11.1)
 3002 FORMAT (50H (M01,M02,MOS,XAM01,X1M01,XAM02,X2M02,XAMOS,X1MOS),
    1                                                      3F8.1,6F8.5)

 3003 FORMAT (33H (ITER,FM7,FM8,L1,L2,LS,E1,E2,ES),I5,2F12.6,6F10.1)
 3004 FORMAT (34H (ITER,FM9,FM10,L1,L2,LS,E1,E2,ES),I5,2F12.6,6F10.1)
```

```
3005 FORMAT (35H (ITER,FM11,FM12,L1,L2,LS,E1,E2,ES),I5,2F12.6,6F10.1)
 2003 FORMAT (36H TOO MANY ITERATIONS ON DY.LT.0 CASE)
3006 FORMAT (35H (ITER,FM11,FM13,L1,L2,LS,E1,E2,ES),I5,2F12.6,6F10.1)
      END
      SUBROUTINE DEFL3(ITYPE)
C
C   DETERMINATION OF DEFLECTION FOR GIVEN LINE LENGTH AND STATION.
C
      REAL L,L1,L2,LS,MO1,MO2,MOS
      COMMON IPRINT,ITER
      COMMON TA,TAS,WO1,WO2,WOS,L,S,DEFL,D,DY,DS,H,DYS,WC,C1,HS1,HS2
      COMMON L1,L2,LS,E1,E2,ES
      COMMON MO1,MO2,MOS,XAMO1,X1MO1,XAMO2,X2MO2,XAMOS,X1MOS
      COMMON WL,SG
      INTEGER OT
      OT=61
      DFINIT=DEFL
      IBO=1
      IBO=0
      IF (IPRINT.EQ.1) IBO=1
      TOL=TA/1000000.0
 1    I=0
C
C   INITIAL ESTIMATE OF DEFLECTION AND PAYLOAD FROM PREVIOUS CASE
      DY=DFINIT*L+S*D
      DYS=DY
      IF (H.GE.0.0) ITYPE=6
      IF (H.LT.0.0) ITYPE=7
      CALL      RCONVG(ITYPE)
      WL1=WL
      DF1=DFINIT
      IF (ABS(WL1).LE.TOL) RETURN
C
C   GET SECOND CATENARY SOLUTION
      DFCHG=0.01
    6 DF2=-WL1/ABS(WL1)*DFCHG+DF1
      IF (DF2.GT.0.0) GO TO 8
      DFCHG=DFCHG*0.5
      GO TO 6
    8 DEFL=DF2
      DY=S*D+L*DEFL
      DYS=DY
      CALL      RCONVG(ITYPE)
      WL2=WL
      IF (ABS(WL2).LE.TOL) RETURN
C
C   ENTER SECANT ITERATION LOOP
   10 RAT=1.0
   11 DF3=DF2-((DF2-DF1)*WL2/(WL2-WL1))*RAT
      IF (DF3.GT.0.0) GO TO 12
      RAT=RAT/2.0
      GO TO 11
   12 DEFL=DF3
      DY=S*D+L*DEFL
      DYS=DY
```

```
      CALL     RCONVG(ITYPE)
      WL3=WL
      IF (IBO.EQ.1) WRITE (OT,2000) I,WL1,WL2,WL3,DF1,DF2,DF3
      IF (ABS(WL3).LE.TOL) RETURN
      I=I+1
      IF (I.GT.20) GO TO 20
      DF1=DF2
      DF2=DF3
      WL1=WL2
      WL2=WL3
      GO TO 10
   20 CONTINUE
      IBO=IBO+1
      IF (IBO.EQ.2) RETURN
      GO TO 1
 2000 FORMAT (21H ITERATIONS IN DEFL3=,I2,6F15.6)
      END

      SUBROUTINE CATARM
C  ROUTINE TO GENERATE CATENARY LENGTHS AND MOMENT ARMS
C
       REAL L,L1,L2,LS,MO1,MO2,MOS
      COMMON IPRINT,ITER
      COMMON TA,TAS,WO1,WO2,WOS,L,S,DEFL,D,DY,DS,H,DYS,WC,C1,HS1,HS2
      COMMON L1,L2,LS,E1,E2,ES
      COMMON MO1,MO2,MOS,XAMO1,X1MO1,XAMO2,X2MO2,XAMOS,X1MOS
      COMMON WL,SG
      L1=MO1*FUNC2(X1MO1,XAMO1)
      L2=MO2*FUNC2(X2MO2,XAMO2)
      LS=MOS*FUNC2(X1MOS,XAMOS)
      E1=MO1*FUNC1(X1MO1,XAMO1)/FUNC2(X1MO1,XAMO1)
      E2=MO2*FUNC1(X2MO2,XAMO2)/FUNC2(X2MO2,XAMO2)
      ES=MOS*FUNC1(X1MOS,XAMOS)/FUNC2(X1MOS,XAMOS)
      RETURN
      END

      SUBROUTINE CATGEO
C ROUTINE TO GENERATE CATENARY GEOMETRIC PARAMETERS- PRODUCTION MODEL

C
       REAL L,L1,L2,LS,MO1,MO2,MOS
      COMMON IPRINT,ITER
      COMMON TA,TAS,WO1,WO2,WOS,L,S,DEFL,D,DY,DS,H,DYS,WC,C1,HS1,HS2
      COMMON L1,L2,LS,E1,E2,ES
      COMMON MO1,MO2,MOS,XAMO1,X1MO1,XAMO2,X2MO2,XAMOS,X1MOS
      COMMON WL,SG
      INTEGER OT
      OT=61
C
 C  ARGUMENT CHECK
 10   IARG=1
      ARG1=D/2.0/MO1
      ARG2=(L-D)/2.0/MO2
      ARG3=DS/2.0/MOS
      IF (ABS(ARG1).GT.87.0) GO TO 90
      IF (ABS(ARG2).GT.87.0) GO TO 90
C  HERE IS WHERE THE PRODUCTION MODEL DIFFERS FROM THE STANDARD
      IF (ABS(ARG3).GT.87.0) MOS=(DS/2.0)/(ARG3/ABS(ARG3)*87.0)
C
```

```
 21    X1MO1=FUNC3(DY,MO1,D)
 22    XAMO1=FUNC4(DY,MO1,D)
 23    X2MO2=FUNC3(DY-H,MO2,L-D)
 24    XAMO2=FUNC4(DY-H,MO2,L-D)
C   THIS MANUEVER IS TO AVOID UNDERFLOW IN THE CALCULATIONS
       IF (ABS(ARG3).GT.20.0) GO TO 30
 25    X1MOS=FUNC3(DYS,MOS,DS)
 26    XAMOS=FUNC4(DYS,MOS,DS)
       GO TO 40
 30    X1MOS=DYS/2.0/MOS
       XAMOS=-DYS/2.0/MOS
C   ARGUMENT CHECK
 40    IARG=2
       IF (ABS(XAMO1).GT.87.0) GOTO 90
       IF (ABS(X1MO1).GT.87.0) GOTO 90
       IF (ABS(XAMO2).GT.87.0) GOTO 90
       IF (ABS(X2MO2).GT.87.0) GOTO 90
       IF (ABS(XAMOS).GT.87.0) GOTO 90
       IF (ABS(X1MOS).GT.87.0) GOTO 90
       RETURN
 90    WRITE (OT,2000) IARG,ARG1,ARG2,ARG3,X1MO1,XAMO1,X2MO2,XAMO2,
     1                                                    X1MOS,XAMOS
       RETURN
 2000  FORMAT (20H ARG LARGE IN CATGEO,I4,9F10.2)
       END

       FUNCTION ASH(X)
       ASH    = ALOG(X+SQRT(X*X+1.0))
       RETURN
       END

       FUNCTION SINH(X)
       SINH    =0.5*(EXP(X)-EXP(-X))
       RETURN
       END

       FUNCTION  COSH(X)
       COSH    = 0.5*(EXP(X) +EXP(-X))
       RETURN
       END

       FUNCTION  FUNC3(X,Y,Z)
       FUNC3         =ASH(X/(2.0*Y*SINH(Z/2.0/Y))) +Z/2.0/Y
       RETURN
       END

       FUNCTION  FUNC4(X,Y,Z)
       FUNC4         =ASH(X/(2.0*Y*SINH(Z/2.0/Y))) -Z/2.0/Y
       RETURN
       END

       FUNCTION  FUNC1(X,Y)
       FUNC1     =(X*SINH(X)-COSH(X)-Y*SINH(X)+COSH(Y))
       RETURN
       END

       FUNCTION  FUNC2(X,Y)
       FUNC2     =(SINH(X)-SINH(Y))
       RETURN
       END
```


U.S. GOVERNMENT PRINTING OFFICE: 1973—787-964 /26 REGION 10

Carson, Ward W., and Donald D. Studier
1973. A computer program for determining the load-carrying capability of the running skyline. USDA For. Serv. Res. Pap. PNW-157, 26 p., illus. Pacific Northwest Forest and Range Experiment Station, Portland, Oregon.

This paper presents a digital computer program to determine running skyline load-carrying capabilities to aid the logging layout designer with this planning task.

KEYWORDS: Logging, computer program, forest cutting systems.

Carson, Ward W., and Donald D. Studier
1973. A computer program for determining the load-carrying capability of the running skyline. USDA For. Serv. Res. Pap. PNW-157, 26 p., illus. Pacific Northwest Forest and Range Experiment Station, Portland, Oregon.

This paper presents a digital computer program to determine running skyline load-carrying capabilities to aid the logging layout designer with this planning task.

KEYWORDS: Logging, computer program, forest cutting systems.

Carson, Ward W., and Donald D. Studier
1973. A computer program for determining the load-carrying capability of the running skyline. USDA For. Serv. Res. Pap. PNW-157, 26 p., illus. Pacific Northwest Forest and Range Experiment Station, Portland, Oregon.

This paper presents a digital computer program to determine running skyline load-carrying capabilities to aid the logging layout designer with this planning task.

KEYWORDS: Logging, computer program, forest cutting systems.

Carson, Ward W., and Donald D. Studier
1973. A computer program for determining the load-carrying capability of the running skyline. USDA For. Serv. Res. Pap. PNW-157, 26 p., illus. Pacific Northwest Forest and Range Experiment Station, Portland, Oregon.

This paper presents a digital computer program to determine running skyline load-carrying capabilities to aid the logging layout designer with this planning task.

KEYWORDS: Logging, computer program, forest cutting systems.

The mission of the PACIFIC NORTHWEST FOREST AND RANGE EXPERIMENT STATION is to provide the knowledge, technology, and alternatives for present and future protection, management, and use of forest, range, and related environments.

Within this overall mission, the Station conducts and stimulates research to facilitate and to accelerate progress toward the following goals:

1. Providing safe and efficient technology for inventory, protection, and use of resources.
2. Development and evaluation of alternative methods and levels of resource management.
3. Achievement of optimum sustained resource productivity consistent with maintaining a high quality forest environment.

The area of research encompasses Oregon, Washington, Alaska, and, in some cases, California, Hawaii, the Western States, and the Nation. Results of the research will be made available promptly. Project headquarters are at:

Fairbanks, Alaska	Portland, Oregon
Juneau, Alaska	Olympia, Washington
Bend, Oregon	Seattle, Washington
Corvallis, Oregon	Wenatchee, Washington
La Grande, Oregon	

Mailing address: Pacific Northwest Forest and Range Experiment Station
P.O. Box 3141
Portland, Oregon 97208

The FOREST SERVICE of the U.S. Department of Agriculture is dedicated to the principle of multiple use management of the Nation's forest resources for sustained yields of wood, water, forage, wildlife, and recreation. Through forestry research, cooperation with the States and private forest owners, and management of the National Forests and National Grasslands, it strives – as directed by Congress – to provide increasingly greater service to a growing Nation.